Beyond Nash Equilibrium: Solution Concepts for the 21st Century*

Joseph Y. Halpern
Cornell University
Dept. of Computer Science
Ithaca, NY 14853
halpern@cs.cornell.edu
http://www.cs.cornell.edu/home/halpern

Abstract

Nash equilibrium is the most commonly-used notion of equilibrium in game theory. However, it suffers from numerous problems. Some are well known in the game theory community; for example, the Nash equilibrium of repeated prisoner's dilemma is neither normatively nor descriptively reasonable. However, new problems arise when considering Nash equilibrium from a computer science perspective: for example, Nash equilibrium is not robust (it does not tolerate "faulty" or "unexpected" behavior), it does not deal with coalitions, it does not take computation cost into account, and it does not deal with cases where players are not aware of all aspects of the game. Solution concepts that try to address these shortcomings of Nash equilibrium are discussed.

1 Introduction

Nash equilibrium is the most commonly-used notion of equilibrium in game theory. Intuitively, a Nash equilibrium is a *strategy profile* (a collection of strategies, one for each player in the game) such that no player can do better by deviating. The intuition behind Nash equilibrium is that it represent a possible steady state of play. It is a fixed point where each player holds correct beliefs about what other players are doing, and plays a best response to those beliefs. Part of what makes Nash equilibrium so attractive is that in games where each player has only finitely many possible deterministic strategies, and we allow mixed (i.e., randomized) strategies, there is guaranteed to be a Nash equilibrium [Nash 1950] (this was, in fact, the key result of Nash's thesis).

For quite a few games, thinking in terms of Nash equilibrium gives insight into what people do (there is a reason that game theory is taught in business schools!). However, as is well known, Nash equilibrium suffers from numerous problems. For example, the Nash equilibrium in games such as repeated prisoner's dilemma is to always defect (see Section 3 for more discussion of repeated prisoner's dilemma). It is hard to make a case that rational players "should" play the Nash equilibrium in this game when "irrational" players who cooperate for a while do much better! Moreover, in a game that is only played once, why should a Nash equilibrium arise when there are multiple Nash equilibria? Players have no way of knowing which one will be played. And even in games where there is a unique Nash equilibrium (like repeated prisoner's dilemma), how do players obtain correct beliefs about what other players are doing if the game is played only once? (See [Kreps 1990] for a discussion of some of these problems.)

Not surprisingly, there has been a great deal of work in the economics community on developing alternative solution concepts. Various alternatives to and refinements of Nash equilibrium have been introduced, including, among many others, *rationalizability*, *sequential equilibrium*, *(trembling hand) perfect equilibrium*, *proper equilibrium*, and *iterated deletion of weakly dominated strategies*. (These notions are discussed in standard game theory text, such as [Fudenberg and Tirole 1991; Osborne and Rubinstein 1994].) Despite some successes, none of these alternative solution concepts address the following three problems with Nash equilibrium, all inspired by computer science concerns.

- Although both computer science and distributed computing are concerned with multiple agents interacting, the focus in the game theory literature has been on the strategic concerns of agents—rational players choosing strategies that are best responses to strategies chosen by other player, the focus in distributed computing has been on problems such as fault tolerance and asynchrony, leading to, for example work on Byzantine agreement [Fischer, Lynch, and Paterson 1985; Pease, Shostak, and Lamport 1980]. Nash equilibrium does not deal with "faulty" or "unexpected" behavior, nor does it deal with colluding agents. In large games, we should expect both.
- Nash equilibrium does not take computational concerns into account. We need solution concepts that can deal with resource-bounded players, concerns that are at the heart of cryptography.
- Nash equilibrium presumes that players have common knowledge of the structure of the game, including all the possible moves that can be made in every situation and all the players in game. This is not always reasonable in, for example, the large auctions played over the internet.

In the following sections, I discuss each of these issues in more detail, and sketch solution concepts that can deal with

*Reprinted from the *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing*, 2008. Work supported in part by NSF under under grants ITR-0325453 and IIS-0534064, and by AFOSR under grant FA9550-05-1-0055.

them, with pointers to the relevant literature.

2 Robust and Resilient Equilibrium

Nash equilibrium tolerates deviations by one player. It is perfectly consistent with Nash equilibrium that two players could do much better by deviating in a coordinated way. For example, consider a game with $n > 1$ players where players much play either 0 or 1. If everyone plays 0, everyone get a payoff of 1; if exactly two players plays 1 and the rest play 0, then the two who play 1 get a payoff of 2, and the rest get 0; otherwise, everyone gets 0. Clearly everyone playing 0 is a Nash equilibrium, but any pair of players can do better by deviating and playing 1.

Say that a Nash equilibrium is k-*resilient* if it tolerates deviations by coalitions of up to k players. The notion of resilience is an old one in the game theory literature, going back to Aumann [1959]. Various extensions of Nash equilibrium have been proposed in the game theory literature to deal with coalitions [Bernheim, Peleg, and Whinston 1989; Moreno and Wooders 1996]. However, these notions do not deal with players who act in unexpected ways.

There can be many reasons that players act in unexpected ways. One, of course, is that they are indeed irrational. However, often seemingly irrational behavior can be explained by players having unexpected utilities. For example, in a peer-to-peer network like Kazaa or Gnutella, it would seem that no rational agent should share files. Whether or not you can get a file depends only on whether other people share files. Moreover, there are disincentives for sharing (the possibility of lawsuits, use of bandwidth, etc.). Nevertheless, people do share files. However, studies of the Gnutella network have shown that almost 70 percent of users share no files and nearly 50 percent of responses are from the top 1 percent of sharing hosts [Adar and Huberman 2000]. Is the behavior of the sharing hosts irrational? It is if we assume appropriate utilities. But perhaps sharing hosts get a big kick out of being the ones that provide everyone else with the music they play. Is that so irrational? In other cases, seemingly irrational behavior can be explained by faulty computers or a faulty network (this, of course, is the concern that work on Byzantine agreement is trying to address), or a lack of understanding of the game.

To give just one example of a stylized game where this issue might be relevant, consider a group of n bargaining agents. If they all stay and bargain, then all get 2. However, if any agent leaves the bargaining table, those who leave get 1, while those who stay get 0. Clearly everyone staying at the bargaining table is a k-resilient Nash equilibrium for all $k \geq 0$, and it is Pareto optimal (everyone in fact gets the highest possible payoff). But, especially if n is large, this equilibrium is rather "fragile"; all it takes is one person to leave the bargaining table for those who stay to get 0.

Whatever the reason, as pointed out by Abraham et al. [2006], it seems important to design strategies that tolerate such unanticipated behaviors, so that the payoffs of the users with "standard" utilities do not get affected by the nonstandard players using different strategies. This can be viewed as a way of adding fault tolerance to equilibrium notions. To capture this intuition, Abraham et al. [Abraham, Dolev, Gonen, and Halpern 2006] define a strategy profile to be t-*immune* if no player who does *not* deviate is worse off if up to t players do deviate. Note the difference between resilience and immunity. A strategy profile is resilient if deviators do not gain by deviating; a profile is immune if non-deviators do not get hurt by deviators. In the example above, although everyone bargaining is a k-resilient Nash equilibrium for all $k \geq 0$, it is not 1-immune.

Of course, we may want to combine resilience and resilience; a strategy is (k, t)-*robust* if it is both k-resilient and t-immune. (All the informal definitions here are completely formalized in [Abraham, Dolev, Gonen, and Halpern 2006; Abraham, Dolev, and Halpern 2008].) A Nash equilibrium is just a (1,0)-robust equilibrium. Unfortunately, for $(k, t) \neq (1, 0)$, a (k, t)-robust equilibrium does not exist in general. Nevertheless, there are a number of games of interest where they do exist; in particular, they can exist if players can take advantage of a *mediator*, or trusted third party. To take just one example, consider Byzantine agreement [Pease, Shostak, and Lamport 1980]. Recall that in Byzantine agreement there are n soldiers, up to t of which may be faulty (the t stands for *traitor*), one of which is the general. The general has an initial preference to attack or retreat. We want a protocol that guarantees that (1) all *nonfaulty* soldiers reach the same decision, and (2) if the general is nonfaulty, then the decision is the general's preference. It is trivial to solve Byzantine agreement with a mediator: the general simply sends the mediator his preference, and the mediator sends it to all the soldiers.

The obvious question of interest is whether we can *implement* the mediator. That is, can the players in the system, just talking among themselves (using what economists call "cheap talk") simulate the effects of the mediator. This is a question that has been of interest to both the computer science community and the game theory community. In game theory, the focus has been on whether a Nash equilibrium in a game with a mediator can be implemented using cheap talk (cf. [Barany 1992; Ben-Porath 2003; Forges 1990; Gerardi 2004; Heller 2005; Urbano and Vila 2002; Urbano and Vila 2004]). In cryptography, the focus has been on *secure multiparty computation* [Goldreich et al. 1987; Shamir et al. 1981; Yao 1982]. Here it is assumed that each agent i has some private information x_i (such private information, like the general's preference, is typically called the player's *type* in game theory). Fix a function f. The goal is have agent i learn $f(x_1, \ldots, x_n)$ without learning anything about x_j for $j \neq i$ beyond what is revealed by the value of $f(x_1, \ldots, x_n)$. With a trusted mediator, this is trivial: each agent i just gives the mediator its private value x_i; the mediator then sends each agent i the value $f(x_1, \ldots, x_n)$. Work on multiparty computation provides general conditions under which this can be done (see [Goldreich 2004] for an overview). Somewhat surprisingly, despite there being over 20 years of work on this problem in both computer science and game theory, until recently, there has been no interaction between the communities on this topic.

Abraham et al. [2006, 2008] essentially characterize when mediators can be implemented. To understand the results, three games need to be considered: an *underlying game* Γ,

an extension Γ_d of Γ with a mediator, and a cheap-talk extension Γ_{CT} of Γ. Γ is assumed to be a *normal-form Bayesian game*: each player has a type from some type space with a known distribution over types, and must choose an action (where the choice can depend on his type). The utilities of the players depend on the types and actions taken. For example, in Byzantine agreement, the possible types of the general are 0 and 1, his possible initial preferences (the types of the other players are irrelevant). The players' actions are to attack or retreat. The assumption that there is a distribution over the general's preferences is standard in game theory, although not so much in distributed computing. Nonetheless, in many applications of Byzantine agreement, it seems reasonable to assume such a distribution. Roughly speaking, a cheap talk game *implements* a game with a mediator if it induces the same distribution over actions in the underlying game, for each type vector of the players. With this background, I can summarize the results of Abraham et al.

- If $n > 3k + 3t$, a (k, t)-robust strategy $\vec{\sigma}$ with a mediator can be implemented using cheap talk (that is, there is a (k, t)-robust strategy $\vec{\sigma}'$ in the cheap talk game such that $\vec{\sigma}$ and $\vec{\sigma}'$ induce the same distribution over actions in the underlying game). Moreover, the implementation requires no knowledge of other agents' utilities, and the cheap talk protocol has bounded running time that does not depend on the utilities.
- If $n \leq 3k + 3t$ then, in general, mediators cannot be implemented using cheap talk without knowledge of other agents' utilities. Moreover, even if other agents' utilities are known, mediators cannot, in general, be implemented without having a $(k + t)$-punishment strategy (that is, a strategy that, if used by all but at most $k + t$ players, guarantees that every player gets a worse outcome than they do with the equilibrium strategy) nor with bounded running time.
- If $n > 2k + 3t$, then mediators can be implemented using cheap talk if there is a punishment strategy (and utilities are known) in finite expected running time that does not depend on the utilities.
- If $n \leq 2k + 3t$ then mediators cannot, in general, be implemented, even if there is a punishment strategy and utilities are known.
- If $n > 2k + 2t$ and there are broadcast channels then, for all ϵ, mediators can be ϵ-implemented (intuitively, there is an implementation where players get utility within ϵ of what they could get by deviating) using cheap talk, with bounded expected running time that does not depend on the utilities.
- If $n \leq 2k + 2t$ then mediators cannot, in general, be ϵ-implemented, even with broadcast channels. Moreover, even assuming cryptography and polynomially-bounded players, the expected running time of an implementation depends on the utility functions of the players and ϵ.
- If $n > k + 3t$ then, assuming cryptography and polynomially-bounded players, mediators can be ϵ-implemented using cheap talk, but if $n \leq 2k + 2t$, then the running time depends on the utilities in the game and ϵ.
- If $n \leq k + 3t$, then even assuming cryptography, polynomially-bounded players, and a $(k + t)$-punishment strategy, mediators cannot, in general, be ϵ-implemented using cheap talk.
- If $n > k + t$ then, assuming cryptography, polynomially-bounded players, and a public-key infrastructure (PKI), we can ϵ-implement a mediator.

All the possibility results showing that mediators can be implemented use techniques from secure multiparty computation. The results showing that that if $n \leq 3k + 3t$, then we cannot implement a mediator without knowing utilities and that, even if utilities are known, a punishment strategy is required, use the fact that Byzantine agreement cannot be reached if $t < n/3$; the impossibility result for $n \leq 2k + 3t$ also uses a variant of Byzantine agreement. These results provide an excellent illustration of how the interaction between computer science and game theory can lead to fruitful insights. Related work on implementing mediators can be found in [Gordon and Katz 2006; Halpern and Teague 2004; Izmalkov, Micali, and Lepinski 2005; Kol and Naor 2008; Lepinski, Micali, Peikert, and Shelat 2004; Lysyanskaya and Triandopoulos 2006].

3 Taking Computation Into Account

Nash equilibrium does not take computation into account. To see why this might be a problem, consider the following example, taken from [Halpern and Pass 2008].

Example 3.1: You are given a number n-bit number x. You can guess whether it is prime, or play safe and say nothing. If you guess right, you get \$10; if you guess wrong, you lose \$10; if you play safe, you get \$1. There is only one Nash equilibrium in this 1-player game: giving the right answer. But if n is large, this is almost certainly not what people will do. Even though primality testing can be done in polynomial time, the costs for doing so (buying a larger computer, for example, or writing an appropriate program), will probably not be worth it for most people. The point here is that Nash equilibrium is not taking the cost of computing whether x is prime into account.

There have been attempts in the game theory community to define solution concepts that take computation into account, going back to the work of Rubinstein [1986]. (See [Kalai 1990] for an overview of the work in this area in the 1980s, and [Ben-Sasson, Kalai, and Kalai 2007] for more recent work.) Rubinstein assumed that players choose a finite automaton to play the game rather than choosing a strategy directly; a player's utility depends both on the move made by the automaton and the complexity of the automaton (identified with the number of states of the automaton). Intuitively, automata that use more states are seen as representing more complicated procedures. Rafael Pass and I [2008] provide a general game-theoretic framework that takes computation into account. (All the discussion in this section is taken from [Halpern and Pass 2008].) Like Rubinstein, we view all players as choosing a machine, but we use Turing machines,

rather than finite automata. We associate a complexity, not just with a machine, but with the machine and its input. This is important in Example 3.1, where the complexity of computing whether x is prime depends, in general, on the length of x.

The complexity could represent the running time of or space used by the machine on that input. The complexity can also be used to capture the complexity of the machine itself (e.g., the number of states, as in Rubinstein's case) or to model the cost of searching for a new strategy to replace one that the player already has. (One of the reasons that players follow a recommended strategy is that there may be too much effort involved in trying to find a new one; I return to this point later.)

We again consider Bayesian games, where each player has a type. In a standard Bayesian game, an agent's utility depends on the type profile and the action profile (that is, every player's type, and the action chosen by each player). In a *computational Bayesian game*, each player i chooses a Turing machine. Player i's type t_i is taken to be the input to player i's Turing machine M_i. The output of M_i on input t_i is taken to be player i's action. There is also a complexity associated with the pair (M_i, t_i). Player i's utility again depends on the type profile and the action profile, and also on the complexity profile. The reason we consider the whole complexity profile in determining player i's utility, as opposed to just i's complexity, is that, for example, i might be happy as long as his machine takes fewer steps than j's. Given these definitions, we can define Nash equilibrium as usual. With this definition, by defining the complexity appropriately, it will be the case that playing safe for sufficiently large inputs will be an equilibrium.

Computational Nash equilibrium also gives a plausible explanation of observed behavior in finitely-repeated prisoner's dilemma.

Example 3.2: Recall that prisoner's dilemma, in prisoner's dilemma, there are two prisoners, who can choose to either cooperate or defect. As described in the table below, if they both cooperate, they both get 3; if they both defect, then both get 1; if one defects and the other cooperates, the defector gets 5 and the cooperator gets -5. (Intuitively, the cooperator stays silent, while the defector "rats out" his partner. If they both rat each other out, they both go to jail.)

	C	D
C	(3,3)	$(-5, 5)$
D	$(5, -5)$	(-3,-3)

It is easy to see that defecting dominates cooperating: no matter what the other player does, a player is better off defecting than cooperating. Thus, "rational" players should defect. And, indeed, (D, D) is the only Nash equilibrium of this game. Although (C, C) gives both players a better payoff than (D, D), this is not an equilibrium.

Now consider finitely repeated prisoner's dilemma (FRPD), where prisoner's dilemma is played for some fixed number N of rounds. The only Nash equilibrium is to always defect; this can be seen by a backwards induction argument. (The last round is like the one-shot game, so both players should defect; given that they are both defecting at the last round, they should both defect at the second-last round; and so on.) This seems quite unreasonable. And, indeed, in experiments, people do not always defect. In fact, quite often they cooperate throughout the game. Are they irrational? It is hard to call this irrational behavior, given that the "irrational" players do much better than supposedly rational players who always defect. There have been many attempts to explain cooperation in FRPD in the literature (see, for example, [Kreps, Milgrom, Roberts, and Wilson 1982]). Indeed, there have even been well-known attempts that take computation into account; it can be shown that if players are restricted to using a finite automaton with bounded complexity, then there exist equilibria that allow for cooperation [Neyman 1985; Papadimitriou and Yannakakis 1994]. However, the strategies used in those equilibria are quite complex, and require the use of large automata; as a consequence this approach does not seem to provide a satisfactory explanation as to why people choose to cooperate.

Using the framework described above leads to a straightforward explanation. Consider the *tit-for-tat* strategy, which proceeds as follows: a player cooperates at the first round, and then at round $m + 1$, does whatever his opponent did at round m. Thus, if the opponent cooperated at the previous round, then you reward him by continuing to cooperate; if he defected at the previous round, you punish him by defecting. If both players play tit-for-tat, then they cooperate throughout the game. Interestingly, tit-for-tat does exceedingly well in FRPD tournaments, where computer programs play each other [Axelrod 1984].

Tit-for-tat is a simple program, which needs very little memory. Suppose that we charge even a modest amount for memory usage, and that there is a discount factor δ, with $.5 < \delta < 1$, so that if the player gets a reward of r_m in round m, his total reward over the whole N-round game is taken to be $\sum_{m=1}^{N} \delta^m r_m$. In this case, it is easy to see that, no matter what the cost of memory is, as long as it is positive, for a sufficiently long game, it will be a Nash equilibrium for both players to play tit-for-tat. For the best response to tit-for-tat is to play tit-for-tat up to the last round, and then to defect. But following this strategy requires the player to keep track of the round number, which requires the use of extra memory. The extra gain of \$2 achieved by defecting at the last round, if sufficiently discounted, will not be worth the cost of keeping track of the round number.

Note that even if only one player is computationally bounded and is charged for memory, and memory is free for the other player, then there is a Nash equilibrium where the bounded player plays tit-for-tat, while the other player plays the best response of cooperating up (but not including) to the round of the game, and then defecting.

Although with standard games there is always a Nash equilibrium, this is not the case when we take computation into account, as the following example shows.

Example 3.3: Consider roshambo (rock-paper-scissors). We model playing rock, paper, and scissors as playing 0, 1, and 2, respectively. The payoff to player 1 of the outcome (i, j)

is 1 if $i = j \oplus 1$ (where $\oplus$ denotes addition mod 3), -1 if $j = i \oplus 1$, and 0 if $i = j$. Player 2's playoffs are the negative of those of player 1; the game is a zero-sum game. As is well known, the unique Nash equilibrium of this game has the players randomizing uniformly between 0, 1, and 2.

Now consider a computational version of roshambo. Suppose that we take the complexity of a deterministic strategy to be 1, and the complexity of a strategy that uses randomization to be 2, and take player i's utility to be his payoff in the underlying Bayesian game minus the complexity of his strategy. Intuitively, programs involving randomization are more complicated than those that do not randomize. With this utility function, it is easy to see that there is no Nash equilibrium. For suppose that (M_1, M_2) is an equilibrium. If M_1 uses randomization, then 1 can do better by playing the deterministic strategy $j \oplus 1$, where j is the action that gets the highest probability according to M_2 (or is the deterministic choice of player 2 if M_2 does not use randomization). Similarly, M_2 cannot use randomization. But it is well known (and easy to check) that there is no equilibrium for roshambo with deterministic strategies.

Is the lack of Nash equilibrium a problem? Perhaps not. Taking computation into account should cause us to rethink things. In particular, we may want to consider other solution concepts. But, as the examples above show, Nash equilibrium does seem to make reasonable predictions in a number of games of interest. Perhaps of even more interest, using computational Nash equilibrium lets us provide a game-theoretic account of security.

The standard framework for multiparty security does not take into account whether players have an incentive to execute the protocol. That is, if there were a trusted mediator, would player i actually use the recommended protocol even if i would be happy to use the services of the mediator to compute the function f? Nor does it take into account whether the adversary has an incentive to undermine the protocol.

Roughly speaking, the game-theoretic definition says that Π is a *game-theoretically secure* (cheap-talk) protocol for computing f if, for all choices of the utility function, if it is a Nash equilibrium to play with the mediator to compute f, then it is also a Nash equilibrium to use Π to compute f. Note that this definition does not mention privacy. It does not need to; this is taken care of by choosing the utilities appropriately. Pass and I [2008] show that, under minimal assumptions, this definition is essentially equivalent to a variant of *zero knowledge* [Goldwasser, Micali, and Rackoff 1989] called *precise zero knowledge* [Micali and Pass 2006]. Thus, the two approaches used for dealing with "deviating" players in two game theory and cryptography—*Nash equilibrium* and *zero-knowledge "simulation"*—are intimately connected; indeed, they are essentially equivalent once we take computation into account appropriately.

4 Taking (Lack of) Awareness Into Account

Standard game theory models implicitly assume that all significant aspects of the game (payoffs, moves available, etc.) are common knowledge among the players. However, this is not always a reasonable assumption. For example, sleazy companies assume that consumers are not aware that they can lodge complaints if there are problems; in a war setting, having technology that an enemy is unaware of (and thus being able to make moves that the enemy is unaware of) can be critical; in financial markets, some investors may not be aware of certain investment strategies (complicated hedging strategies, for example, or tax-avoidance strategies).

To understand the impact of adding the possibility of unawareness to the analysis of games, consider the game shown in Figure 1 (this example, and all the discussion in this section, is taken from [Halpern and Rêgo 2006]). One Nash equilibrium of this game has A playing across$_A$ and B playing down$_B$. However, suppose that A is not aware that B can play down$_B$. In that case, if A is rational, A will play down$_A$. Therefore, Nash equilibrium does not seem to be the appropriate solution concept here. Although A would play across$_A$ if A knew that B were going to play down$_B$, A cannot even contemplate this possibility, let alone know it.

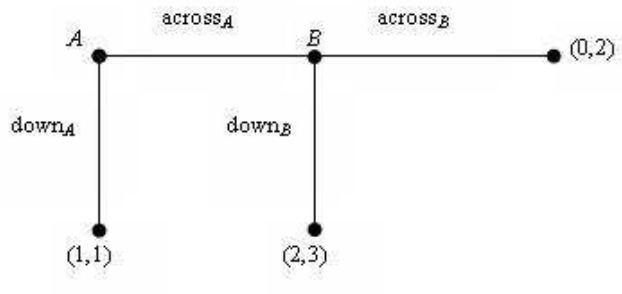

Figure 1: A simple game.

To find an appropriate analogue of Nash equilibrium in games where players may be unaware of some possible moves, we must first find an appropriate representation for such games. The first step in doing so is to explicitly represent what players are aware of at each node. We do this by using what we call an *augmented game*.

Recall that an *extensive game* is described by a *game tree*. Each node in the tree describes a partial *history* of the game—the sequence of moves that led to that node. Associated with each node is the player that moves at that node. Some nodes where a player i moves are grouped together into an *information set for player* i. Intuitively, if player i is at some node in an information set I, then i does not know which node of I describes the true situation; thus, at all nodes in I, i must make the same move. An augmented game is an extensive game with one more feature: associated with each node in the game tree where player i moves is the *level of awareness* of player i—the set of histories that player i is aware of.

We use the player's awareness level as a way of keeping track of how the player's awareness changes over time. For example, perhaps A playing across$_A$ will result in B becoming aware of the possibility of playing down$_B$. In financial settings, one effect of players using certain investment strategies is that other players become aware of the possibil-

ity of using that strategy. Strategic thinking in such games must take this possibility into account. We would model this possibility by having some probability of B's awareness level changing. (The formal definition of augmented game can be found in [Halpern and Rêgo 2006].)

For example, suppose that in the game shown in Figure 1

- players A and B are aware of all histories of the game;
- player A is uncertain as to whether player B is aware of run $\langle \text{across}_A, \text{down}_B \rangle$ and believes that he is unaware of it with probability p; and
- the type of player B that is aware of the run $\langle \text{across}_A, \text{down}_B \rangle$ is aware that player A is aware of all histories, and he knows A is uncertain about his awareness level and knows the probability p.

Because A and B are actually aware of all histories of the underlying game, from the point of view of the modeler, the augmented game is essentially identical to the game described in Figure 1, with the awareness level of both players A and B consisting of all histories of the underlying game. However, when A moves at the node labeled A in the modeler's game, she believes that the actual augmented game is Γ^A, as described in Figure 2. In Γ^A, nature's initial move captures A's uncertainty about B's awareness level. At the information set labeled $A.1$, A is aware of all the runs of the underlying game. Moreover, at this information set, A believes that the true game is Γ^A.

At the node labeled $B.1$, B is aware of all the runs of the underlying game and believes that the true game is the modeler's game; but at the node labeled $B.2$, B is not aware that he can play down_B, and so believes that the true game is the augmented game Γ^B described in Figure 3. At the nodes labeled $A.3$ and $B.3$ in the game Γ^B, neither A nor B is aware of the move down_B. Moreover, both players think the true game is Γ^B.

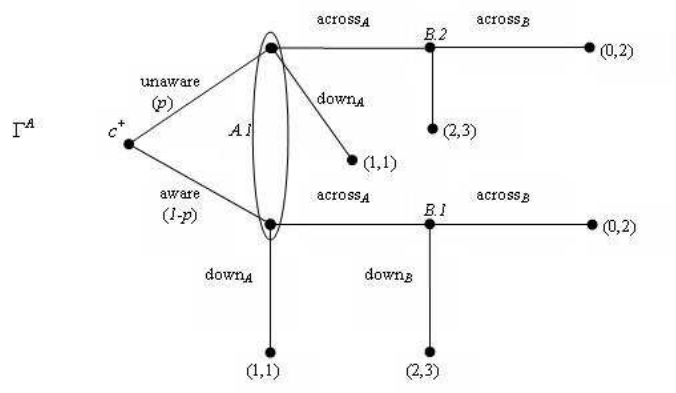

Figure 2: The augmented game Γ^A.

As this example should make clear, to model a game with possibly unaware players, we need to consider, not just one augmented game, but a collection of them. Moreover, we need to describe, at each history in an augmented game, which augmented game the player playing at that history believes is the actual augmented game being played.

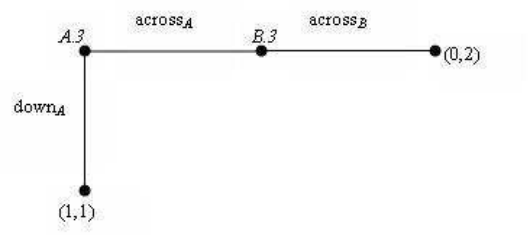

Figure 3: The augmented game Γ^B.

To capture these intuitions, starting with an underlying extensive-form game Γ, we define a *game with awareness based on* Γ to be a tuple $\Gamma^* = (\mathcal{G}, \Gamma^m, \mathcal{F})$, where

- $\mathcal{G}$ is a countable set of augmented games based on Γ, of which one is Γ^m;
- $\mathcal{F}$ maps an augmented game $\Gamma^+ \in \mathcal{G}$ and a history h in Γ^+ such that $P^+(h) = i$ to a pair (Γ^h, I), where $\Gamma^h \in \mathcal{G}$ and I is an information set for player i in game Γ^h.

Intuitively, Γ^m is the game from the point of view of an omniscient modeler. If player i moves at h in game $\Gamma^+ \in \mathcal{G}$ and $\mathcal{F}(\Gamma^+, h) = (\Gamma^h, I)$, then Γ^h is the game that i believes to be the true game when the history is h, and I consists of the set of histories in Γ^h he currently considers possible. For example, in the examples described in Figures 2 and 3, taking Γ^m to the augmented game in Figure 1, we have $\mathcal{F}(\Gamma^m, \langle\,\rangle) = (\Gamma^A, I)$, where I is the information set labeled $A.1$ in Figure 2, and $\mathcal{F}(\Gamma^A, \langle \text{unaware,across}_A \rangle) = (\Gamma^B, \{\langle \text{across}_A \rangle\})$. There are a number of consistency conditions that have to be satisfied by the function $\mathcal{F}$; the details can be found in [Halpern and Rêgo 2006].

The standard notion of Nash equilibrium consists of a profile of strategies, one for each player. Our generalization consists of a profile of strategies, one for each pair (i, Γ'), where Γ' is a game that agent i considers to be the true game in some situation. Intuitively, the strategy for a player i at Γ' is the strategy i would play in situations where i believes that the true game is Γ'. To understand why we may need to consider different strategies consider, for example, the game of Figure 1. B would play differently depending on whether or not he was aware of down_B. Roughly speaking, a profile $\vec{\sigma}$ of strategies, one for each pair (i, Γ'), is a *generalized Nash equilibrium* if $\sigma_{i,\Gamma'}$ is a best response for player i if the true game is Γ', given the strategies $\sigma_{j,\Gamma'}$ being used by the other players in Γ'. As shown in [Halpern and Rêgo 2006], every game with awareness has a generalized Nash equilibrium.

A standard extensive-form game Γ can be viewed as a special case of a game with awareness, by taking $\Gamma^m = \Gamma$, $\mathcal{G} = \{\Gamma^m\}$, and $\mathcal{F}(\Gamma^m, h) = (\Gamma^m, I)$, where I is the information set that contains h. Intuitively, Γ corresponds to the game of awareness where it is common knowledge that Γ is being played. We call this the *canonical representation* of Γ as a game with awareness. It is not hard to show that a strategy profile $\vec{\sigma}$ is a Nash equilibrium of Γ iff it is a generalized Nash equilibrium of the canonical representation of Γ as a game with awareness. Thus, generalized Nash equilibrium can be viewed as a generalization of standard Nash

equilibrium.

Up to now, I have considered only games where players are not aware of their lack of awareness. But in some games, a player might be aware that there are moves that another player (or even she herself) might be able to make, although she is not aware of what they are. Such awareness of unawareness can be quite relevant in practice. For example, in a war setting, even if one side cannot conceive of a new technology available to the enemy, they might believe that there is some move available to the enemy without understanding what that particular move is. This, in turn, may encourage peace overtures. To take another example, an agent might delay making a decision because she considers it possible that she might learn about more possible moves, even if she is not aware of what these moves are.

Although, economists usually interpret awareness as "being able to conceive about an event or a proposition", there are other possible meanings for this concept. For example, awareness may also be interpreted as "understanding the primitive concepts in an event or proposition", or as "being able to determine if an event occurred or not", or as "being able to compute the consequences of some fact" [Fagin and Halpern 1988]. If we interpret "lack of awareness" as "unable to compute" (note that this interpretation is closely related to the discussion of the previous section!), then awareness of unawareness becomes even more significant. Consider a chess game. Although all players understand in principle all the moves that can be made, they are certainly not aware of all consequences of all moves. A more accurate representation of chess would model this computational unawareness explicitly. We provide such a representation.

Roughly speaking, we capture the fact that player i is aware that, at a node h in the game tree, there is a move that j can make she (i) is not aware by having i's subjective representation of the game include a "virtual" move for j at node h. Since i might have only an incomplete understanding of what can happen after this move, i simply describes what she believes will be the game after the virtual move, to the extent that she can. In particular, if she has no idea what will happen after the virtual move, then she can describe her beliefs regarding the payoffs of the game. Thus, our representation can be viewed as a generalization of how chess programs analyze chess games. They explore the game tree up to a certain point, and then evaluate the board position at that point. We can think of the payoffs following a virtual move by j in i's subjective representation of a chess game as describing the evaluation of the board from i's point of view. This seems like a much more reasonable representation of the game than the standard complete game tree!

All the definitions of games with awareness can be generalized to accommodate awareness of unawareness. In particular, we can define a generalized Nash equilibrium as before, and once again show that every game with awareness (including awareness of unawareness) has a generalized Nash equilibrium [Halpern and Rêgo 2006].

There has been a great deal of work recently on modeling unawareness in games. The first papers on the topic was by Feinberg [2004, 2005]. My work with Rêgo [2006] was the first to consider awareness in extensive games, modeling how awareness changed over time. There has been a recent flurry on the topic in the economics literature; see, for example, [Heifetz, Meier, and Schipper 2006b; Li 2006a; Li 2006b; Ozbay 2007]. Closely related is work on logics that include awareness. This work started in the computer science literature [Fagin and Halpern 1988], but more recently, the bulk of the work has appeared in the economics literature (see, for example, [Dekel, Lipman, and Rustichini 1998; Halpern 2001; Halpern and Rêgo 2008; Heifetz, Meier, and Schipper 2006a; Modica and Rustichini 1994; Modica and Rustichini 1999]).

5 Conclusions

I have considered three ways of going beyond standard Nash equilibrium, which take fault tolerance, computation, and lack of awareness into account, respectively. These are clearly only first steps. Here are some directions for further research (some of which I am currently engaged in with my collaborators):

- For example, while (k, t)-robust equilibrium does seem to be a reasonable way of capturing some aspects of robustness, for some applications, it does not go far enough. I said earlier that in economics, all players were assumed to be strategic, or "rational"; in distributed computing, all players were either "good" (and followed the recommended protocol) or "bad" (in which case they could be arbitrarily malicious). Immunity takes into account the bad players. The definition of immunity requires that the rational players are not hurt no matter what the "bad" players do. But this may be too strong. As Ayer et al. [2005] point out, it is reasonable to expect a certain fraction of players in a system to be "good" and follow the recommended protocol, even if it is not a best reply. In general, it may be hard to figure out what the best reply is, so if following the recommended protocol is not unreasonable, they will do that. (Note that this can be captured in a computational model of equilibrium, by charging for switching from the recommended strategy.)

  There may be other standard ways that players act irrational. For example, Kash, Friedman, and I [2007] consider scrip systems, where players perform work in exchange for scrip. There is a Nash equilibrium where everyone uses a *threshold strategy*, performing work only when they have less scrip than some threshold amount. Two standard ways of acting "irrationally" in such a system are to (a) hoard scrip and (b) provide service for free (this is the analogue of posting music on Kazaa). A robust solution should take into account these more standard types of irrational behavior, without perhaps worrying as much about arbitrary irrational behavior.
- The definitions of computational Nash equilibrium considered only Bayesian games. What would appropriate solution concepts be for extensive-form games? Some ideas from the work on awareness seem relevant here, especially if we think of "lack of awareness" as "unable to compute".
- Where do the beliefs come from in an equilibrium with awareness? That is, if I suddenly become aware that you

can make a certain move, what probability should I assign to you making that move? Ozbay [2007] proposes a solution concept where the beliefs are part of the solution concept. He considers only a simple setting, where one player is aware of everything (so that revealing information is purely strategic). Can his ideas be extended to a more general setting?

Agents playing a game can be viewed participating in a concurrent, distributed protocol. Game theory does not take the asynchrony into account, but it can make a big difference. For example, all the results from [Abraham, Dolev, Gonen, and Halpern 2006; Abraham, Dolev, and Halpern 2008] mentioned in Section 2 depend on the system being synchronous. Things are more complicted in asynchronous settings. Getting solution concepts and that deal well with with asynchrony is clearly important.

Another issue that plays a major role in computer science but has thus far not been viewed as significant in game theory, but will, I believe, turn out to be important to the problem of defining appropriate solution concepts, is the analogue of specifying and verifying programs. Games are typically designed to solve certain problems. Thus, for example, economists want to design a spectrum auction so that the equilibrium has certain features. As I pointed out in an earlier overview [Halpern 2003], game theory has typically focused on "small" games: games that are easy to describe, such as Prisoner's Dilemma. The focus has been on subtleties regarding basic issues such as rationality and coordination. To the extent that game theory is used to tackle larger, more practical problems, and especially to the extent that it is computers, or software agents, playing games, rather than people, it will be important to specify carefully exactly what a solution to the game must accomplish. For example, in the context of a spectrum auction, a specification will have to address what should happen if a computer crashes while an agent is in the middle of transmitting a bid, how to deal with agents bidding on slow lines, dealing with agents who win but then go bankrupt, and so on.

Finding logics to reason about solutions, especially doing so in a way that takes into account robustness and asynchrony, seems to me a difficult and worthwhile challenge. Indeed, one desideratum for a good solution concept is that it should be easy to reason about. Pursuing this theme, computer scientists have learned that one good way of designing correct programs is to do so in a modular way. Can a similar idea be applied in game theory? That is, can games designed for solving smaller problems be combined in a seamless way to solve a larger problem. If so, results about *composability of solutions* will be needed; we might want a solution concept that allows for such composability.

References

Abraham, I., D. Dolev, R. Gonen, and J. Halpern (2006). Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proc. 25th ACM Symposium on Principles of Distributed Computing*, pp. 53–62.

Abraham, I., D. Dolev, and J. Halpern (2008). Lower bounds on implementing robust and resilient mediators. In *Fifth Theory of Cryptography Conference*, pp. 302–319.

Adar, E. and B. Huberman (2000). Free riding on Gnutella. *First Monday 5*(10).

Aumann, R. J. (1959). Acceptable points in general cooperative n-person games. In A. Tucker and R. Luce (Eds.), *Contributions to the Theory of Games IV, Annals of Mathematical Studies 40*, pp. 287–324. Princeton University Press.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books.

Ayer, A., L. Alvisi, A. Clement, M. Dahlin, J. Martin, and C. Porth (2005). BAR fault tolerance for cooperative services. In *Proc. 20th ACM Symposium on Operating Systems Principles (SOSP 2005)*, pp. 45–58.

Barany, I. (1992). Fair distribution protocols or how the players replace fortune. *Mathematics of Operations Research 17*, 327–340.

Ben-Porath, E. (2003). Cheap talk in games with incomplete information. *Journal of Economic Theory 108*(1), 45–71.

Ben-Sasson, E., A. Kalai, and E. Kalai (2007). An approach to bounded rationality. In *Advances in Neural Information Processing Systems 19 (Proc. of NIPS 2006)*, pp. 145–152.

Bernheim, B. D., B. Peleg, and M. Whinston (1989). Coalition proof Nash equilibrium: concepts. *Journal of Economic Theory 42*(1), 1–12.

Dekel, E., B. Lipman, and A. Rustichini (1998). Standard state-space models preclude unawareness. *Econometrica 66*, 159–173.

Fagin, R. and J. Y. Halpern (1988). Belief, awareness, and limited reasoning. *Artificial Intelligence 34*, 39–76.

Feinberg, Y. (2004). Subjective reasoning—games with unawareness. Technical Report Resarch Paper Series #1875, Stanford Graduate School of Business.

Feinberg, Y. (2005). Games with incomplete awareness. Technical Report Resarch Paper Series #1894, Stanford Graduate School of Business.

Fischer, M. J., N. A. Lynch, and M. S. Paterson (1985). Impossibility of distributed consensus with one faulty processor. *Journal of the ACM 32*(2), 374–382.

Forges, F. (1990). Universal mechanisms. *Econometrica 58*(6), 1341–64.

Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press.

Gerardi, D. (2004). Unmediated communication in games with complete and incomplete information. *Journal of Economic Theory 114*, 104–131.

Goldreich, O. (2004). *Foundations of Cryptography, Vol. 2*. Cambridge University Press.

Goldreich, O., S. Micali, and A. Wigderson (1987). How to play any mental game. In *Proc. 19th ACM Symposium on Theory of Computing*, pp. 218–229.

Goldwasser, S., S. Micali, and C. Rackoff (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing 18*(1), 186–208.

Gordon, D. and J. Katz (2006). Rational secret sharing, revisited. In *SCN (Security in Communication Networks) 2006*, pp. 229–241.

Halpern, J. Y. (2001). Alternative semantics for unawareness. *Games and Economic Behavior 37*, 321–339.

Halpern, J. Y. (2003). A computer scientist looks at game theory. *Games and Economic Behavior 45*(1), 114–132.

Halpern, J. Y. and R. Pass (2008). Game theory with costly computation. Unpublished manuscript.

Halpern, J. Y. and L. C. Rêgo (2006). Extensive games with possibly unaware players. In *Proc. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 744–751. Full version available at arxiv.org/abs/0704.2014.

Halpern, J. Y. and L. C. Rêgo (2008). Interactive unawareness revisited. *Games and Economic Behavior 62*(1), 232–262.

Halpern, J. Y. and V. Teague (2004). Rational secret sharing and multiparty computation: extended abstract. In *Proc. 36th ACM Symposium on Theory of Computing*, pp. 623–632.

Heifetz, A., M. Meier, and B. Schipper (2006a). Interactive unawareness. *Journal of Economic Theory 130*, 78–94.

Heifetz, A., M. Meier, and B. Schipper (2006b). Unawareness, beliefs and games. Unpublished manuscript, available at www.econ.ucdavis.edu/faculty/schipper/unawprob.pdf.

Heller, Y. (2005). A minority-proof cheap-talk protocol. Unpublished manuscript.

Izmalkov, S., S. Micali, and M. Lepinski (2005). Rational secure computation and ideal mechanism design. In *Proc. 46th IEEE Symposium on Foundations of Computer Science*, pp. 585–595.

Kalai, E. (1990). Bounded rationality and strategic complexity in repeated games. In *Game Theory and Applications*, pp. 131–157. Academic Press.

Kash, I., E. J. Friedman, and J. Y. Halpern (2007). Optimizing scrip systems: efficiency, crashes, hoarders, and altruists. In *Proc. Eighth ACM Conference on Electronic Commerce*, pp. 305–315.

Kol, G. and M. Naor (2008). Cryptography and game theory: Designing protocols for exchanging information. In *Theory of Cryptography Conference*, pp. 320–339.

Kreps, D., P. Milgrom, J. Roberts, and R. Wilson (1982). Rational cooperation in finitely repeated prisoners' dilemma. *Journal of Economic Theory 27*(2), 245–252.

Kreps, D. M. (1990). *Game Theory and Economic Modeling*. Oxford University Press.

Lepinski, M., S. Micali, C. Peikert, and A. Shelat (2004). Completely fair SFE and coalition-safe cheap talk. In *Proc. 23rd ACM Symposium on Principles of Distributed Computing*, pp. 1–10.

Li, J. (2006a). Information structures with unawareness. Unpublished manuscript.

Li, J. (2006b). Modeling unawareness without impossible states. Unpublished manuscript.

Lysyanskaya, A. and N. Triandopoulos (2006). Rationality and adveresarial behavior in multi-party comptuation. In *CRYPTO 2006*, pp. 180–197.

Micali, S. and R. Pass (2006). Local zero knowledge. In *Proc. 38th ACM Symposium on Theory of Computing*, pp. 306–315.

Modica, S. and A. Rustichini (1994). Awareness and partitional information structures. *Theory and Decision 37*, 107–124.

Modica, S. and A. Rustichini (1999). Unawareness and partitional information structures. *Games and Economic Behavior 27*(2), 265–298.

Moreno, D. and J. Wooders (1996). Coalition-proof equilibrium. *Games and Economic Behavior 17*(1), 80–112.

Nash, J. (1950). Equilibrium points in n-person games. *Proc. National Academy of Sciences 36*, 48–49.

Neyman, A. (1985). Bounded complexity justifies cooperation in finitely repated prisoner's dilemma. *Economic Letters 19*, 227–229.

Osborne, M. J. and A. Rubinstein (1994). *A Course in Game Theory*. MIT Press.

Ozbay, E. (2007). Unawareness and strategic announcements in games with uncertainty. In *Theoretical Aspects of Rationality and Knowledge: Proc. Eleventh Conference (TARK 2007)*, pp. 231–238.

Papadimitriou, C. H. and M. Yannakakis (1994). On complexity as bounded rationality. In *Proc. 26th ACM Symposium on Theory of Computing*, pp. 726–733.

Pease, M., R. Shostak, and L. Lamport (1980). Reaching agreement in the presence of faults. *Journal of the ACM 27*(2), 228–234.

Rubinstein, A. (1986). Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory 39*, 83–96.

Shamir, A., R. L. Rivest, and L. Adelman (1981). Mental poker. In D. A. Klarner (Ed.), *The Mathematical Gardner*, pp. 37–43. Prindle, Weber, and Schmidt.

Urbano, A. and J. E. Vila (2002). Computational complexity and communication: Coordination in two-player games. *Econometrica 70*(5), 1893–1927.

Urbano, A. and J. E. Vila (2004). Computationally restricted unmediated talk under incomplete information. *Economic Theory 23*(2), 283–320.

Yao, A. (1982). Protocols for secure computation (extended abstract). In *Proc. 23rd IEEE Symposium on Foundations of Computer Science*, pp. 160–164.